

Non-linear Principal Component Analysis Using Neural Networks

May 23, 2001 and September 30, 2001

Marcin Wolter

Tufts University

HIP Kraków

mwlter@mail.gov

www-cdt.tufts.edu/~mwlter

Marcin Wolter, Non-
linear PCA with NN

Preprocessing – dimensionality reduction

- ◆ Preprocessing – data transformation into a new representation before analyzing them.
- ◆ Dimensionality reduction – can significantly improve the pattern recognition, but as much information as possible should be preserved.
- ◆ Performed typically as an unsupervised learning – only input data are used (no information whether an event originates from signal or from background).

Standard (Linear) Principal Component Analysis

- ◆ Based on rotations in N -dimensional space and replacing M vector coefficients by constants b_i :

$$x \rightarrow \tilde{x} = \sum_{i=1}^{N-M} z_i u_i + \sum_{i=N-M+1}^N b_i u_i$$

where u_i is a new, optimal set of orthonormal vectors.

- ◆ Optimal set — smallest possible difference between original and transformed vectors.

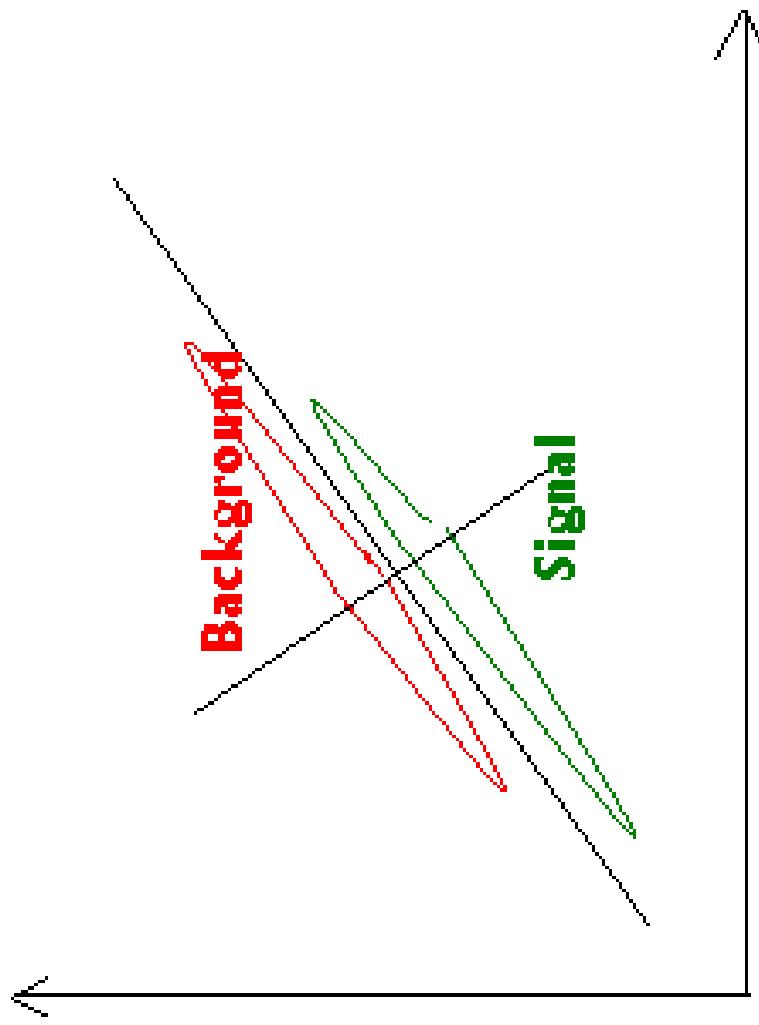
Linear transformation

An example of dimensionality reduction using a linear principal component analysis.

Reduced dimension

When does it fail?

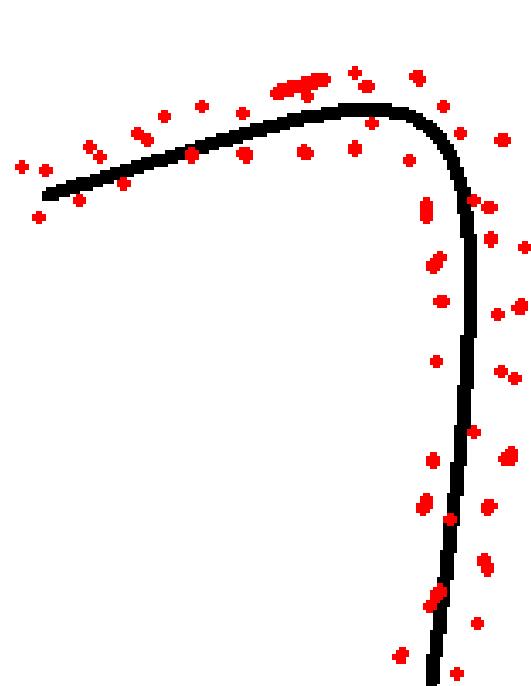
We can't distinguish between signal and background after PCA.



Non-linear Analysis

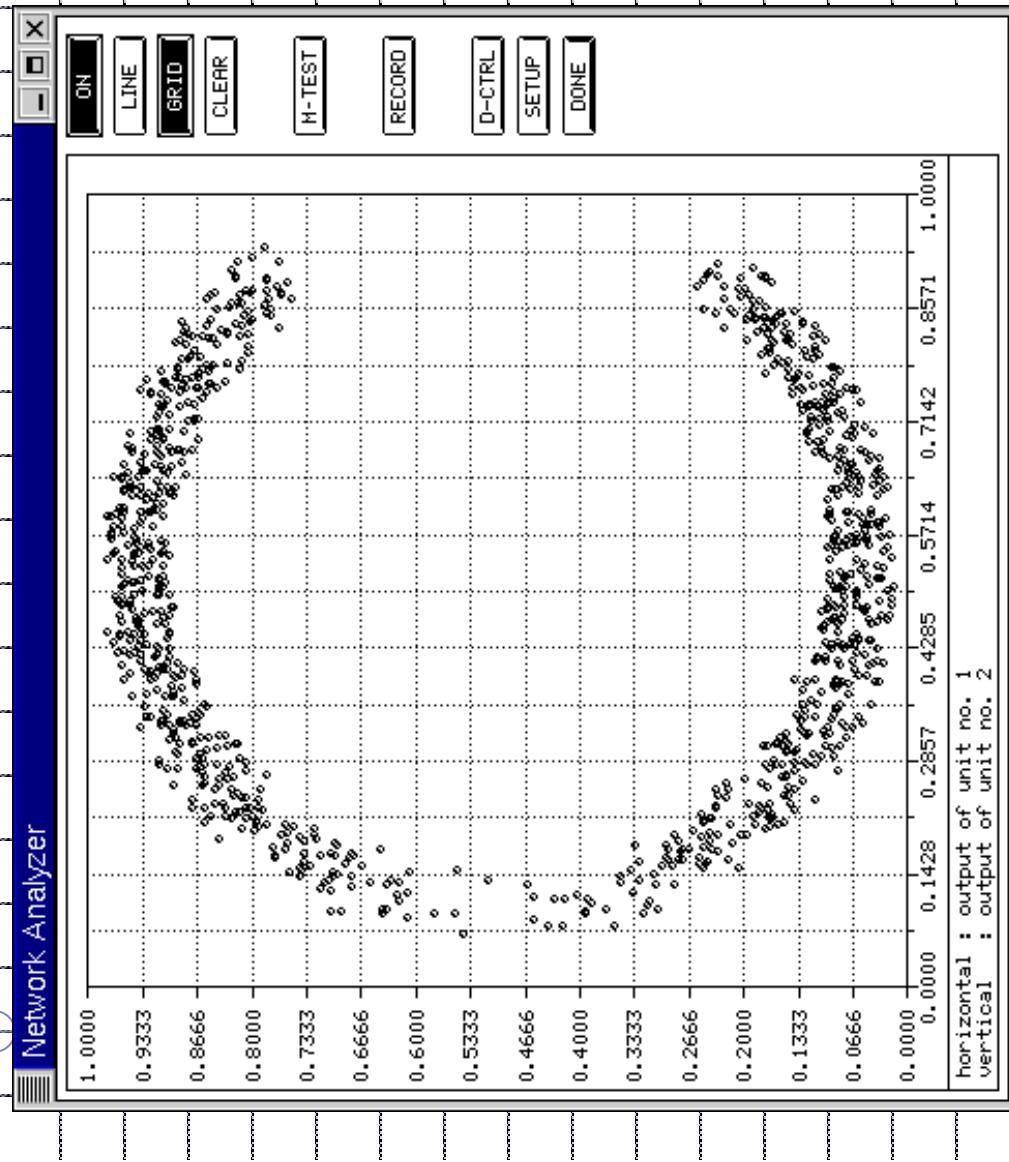
Non-linear transformation
needed.

Neural Network is
non-linear!



March Wolter, Non-
linear PCA with NN

An Example of nonlinear problem.



◆ Intrinsic 1-dimensional.

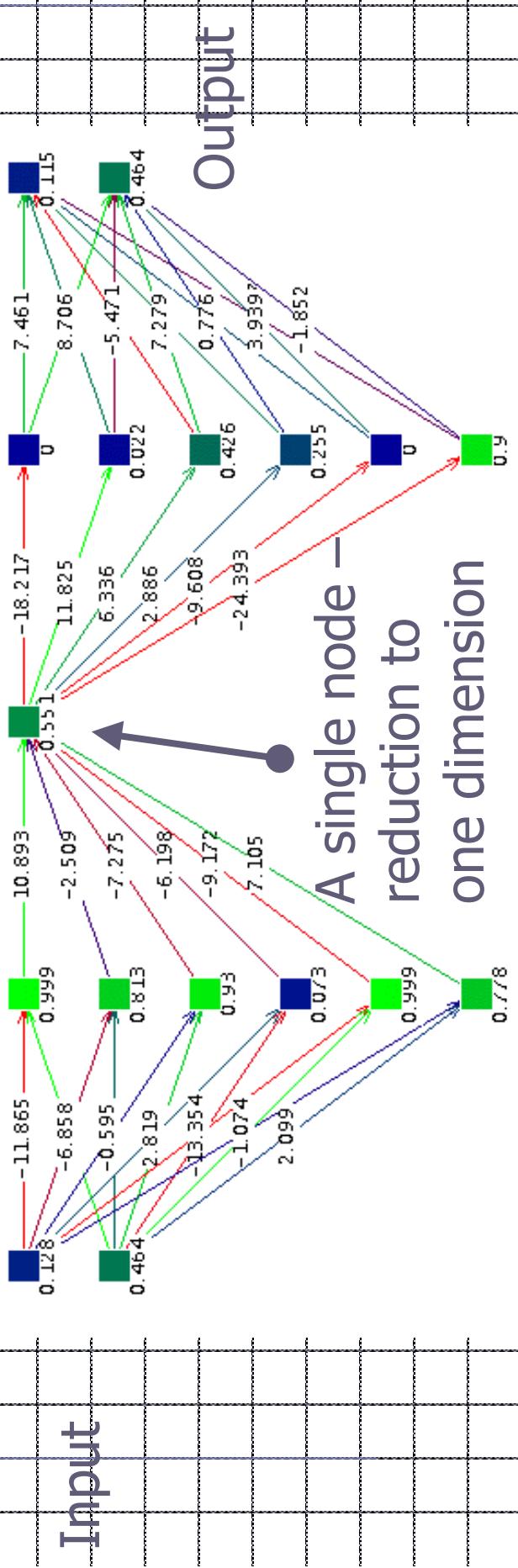
◆ A non-linear problem.

How to map it into 1-dimension?

How to use a Neural Network

- Build a network with two input nodes and two output nodes.
- There should be a hidden layer with **only one** node.
- Train a network giving the same vectors as inputs and outputs.
- At the end the network is trained to reproduce input vectors on output.

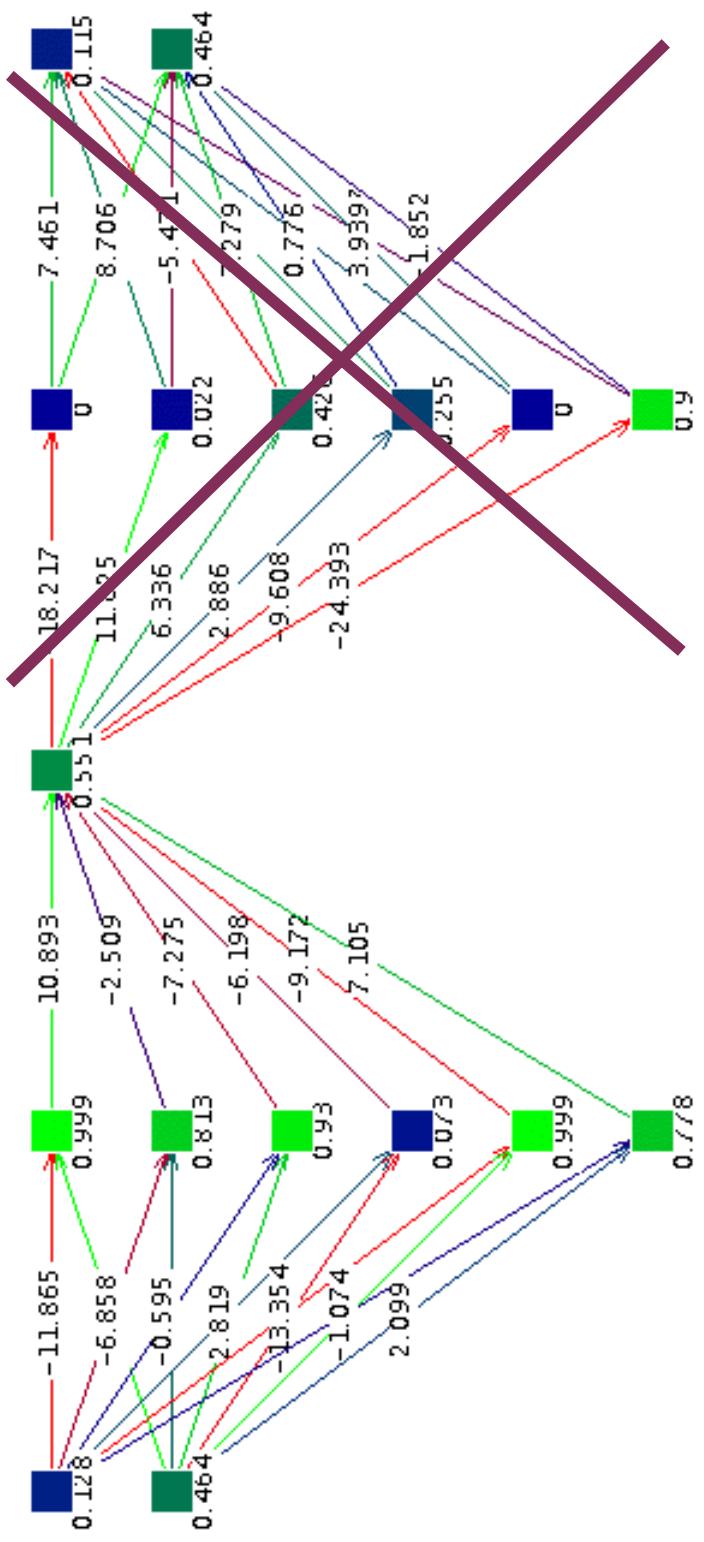
Neural Network for Dimensionality Reduction



A trained neural network

March Wolter, Non-linear PCA with NN

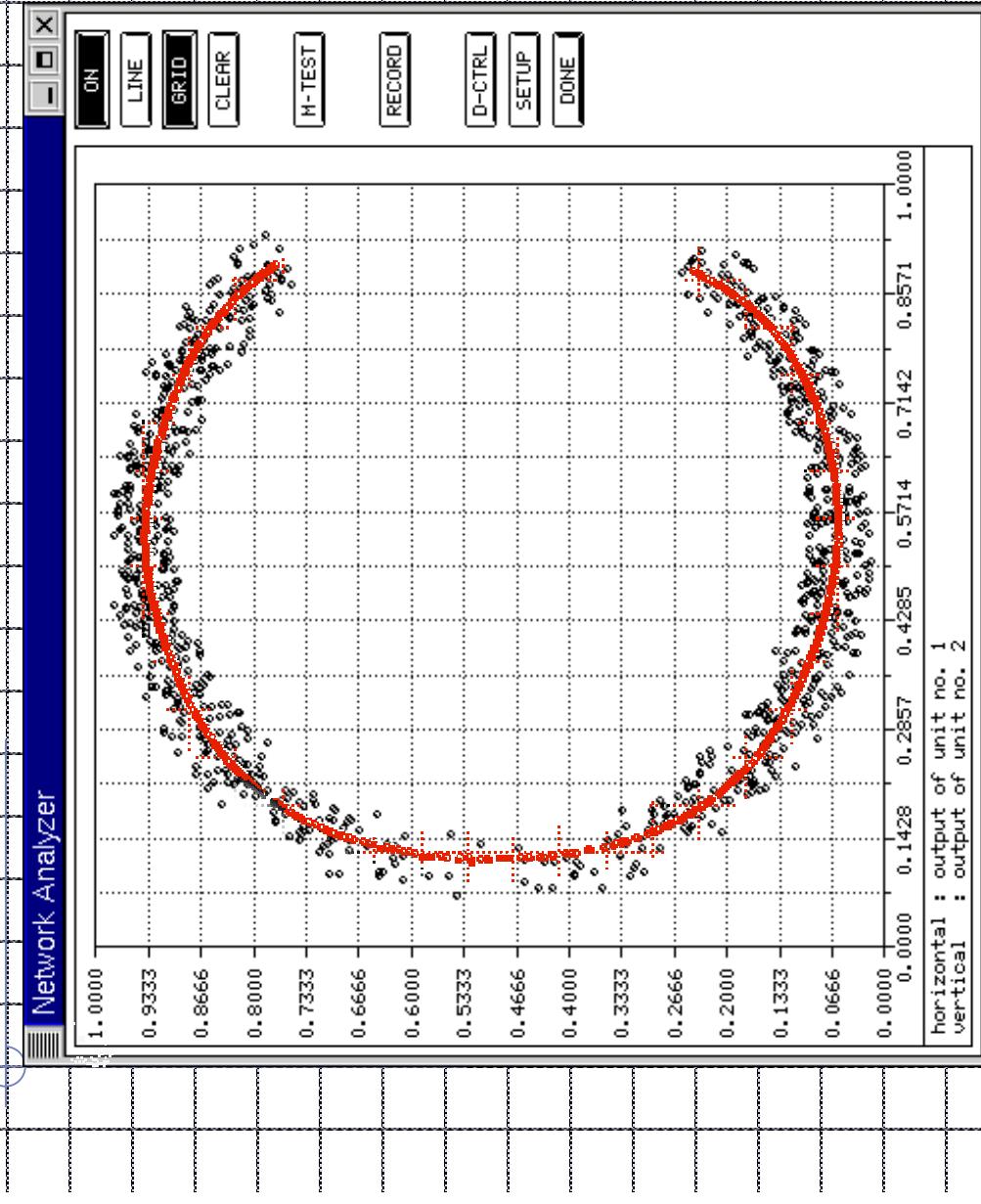
Cutting NN in half



A network transforms from one into two dimensions.

Marcin Wolter, Non-linear PCA with NN

Results – 1-dim pattern

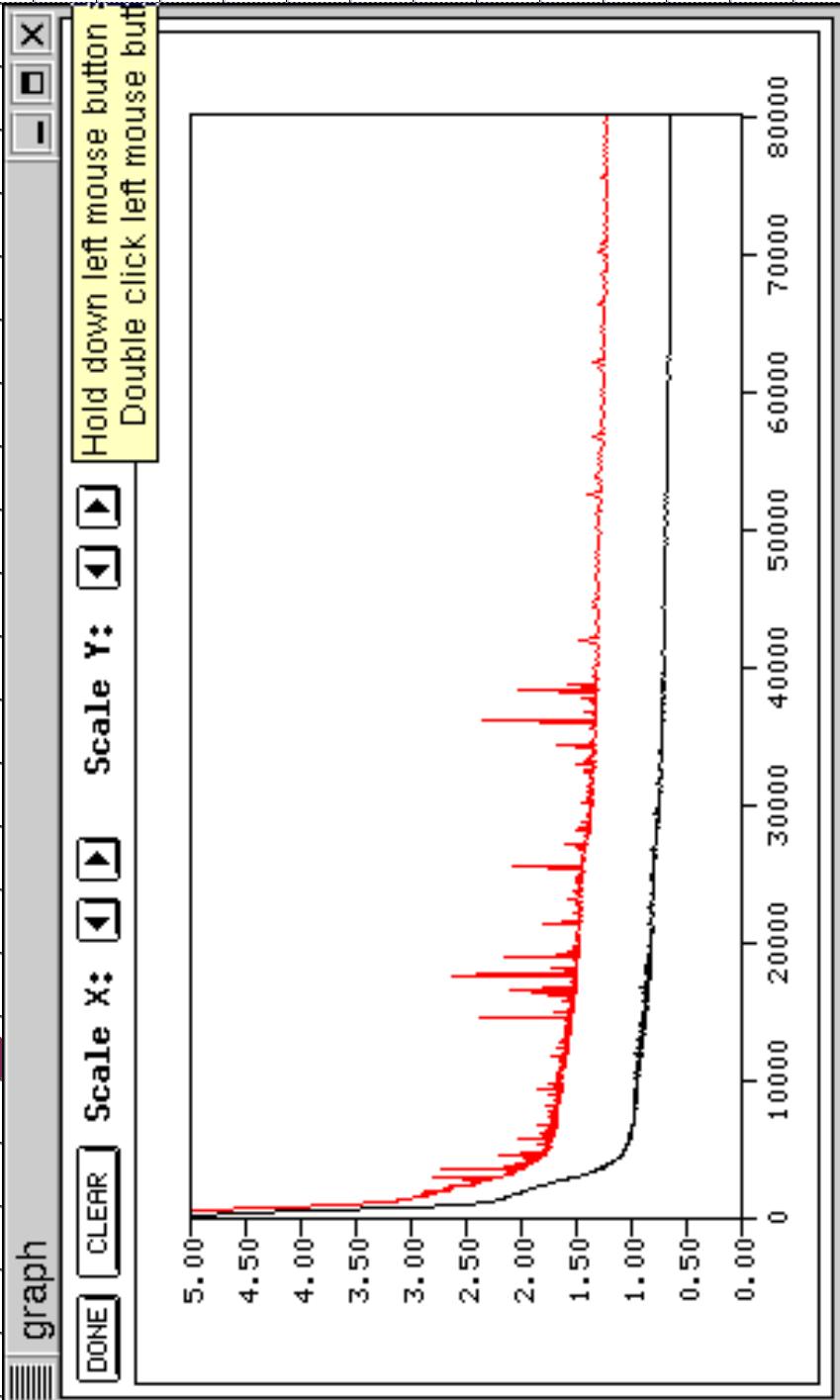


◆ Pattern found
and generalized
(no significant
overlearning).

◆ A non-linear
transformation

Marcin Wójtowicz,
non-
linear PCA with NN

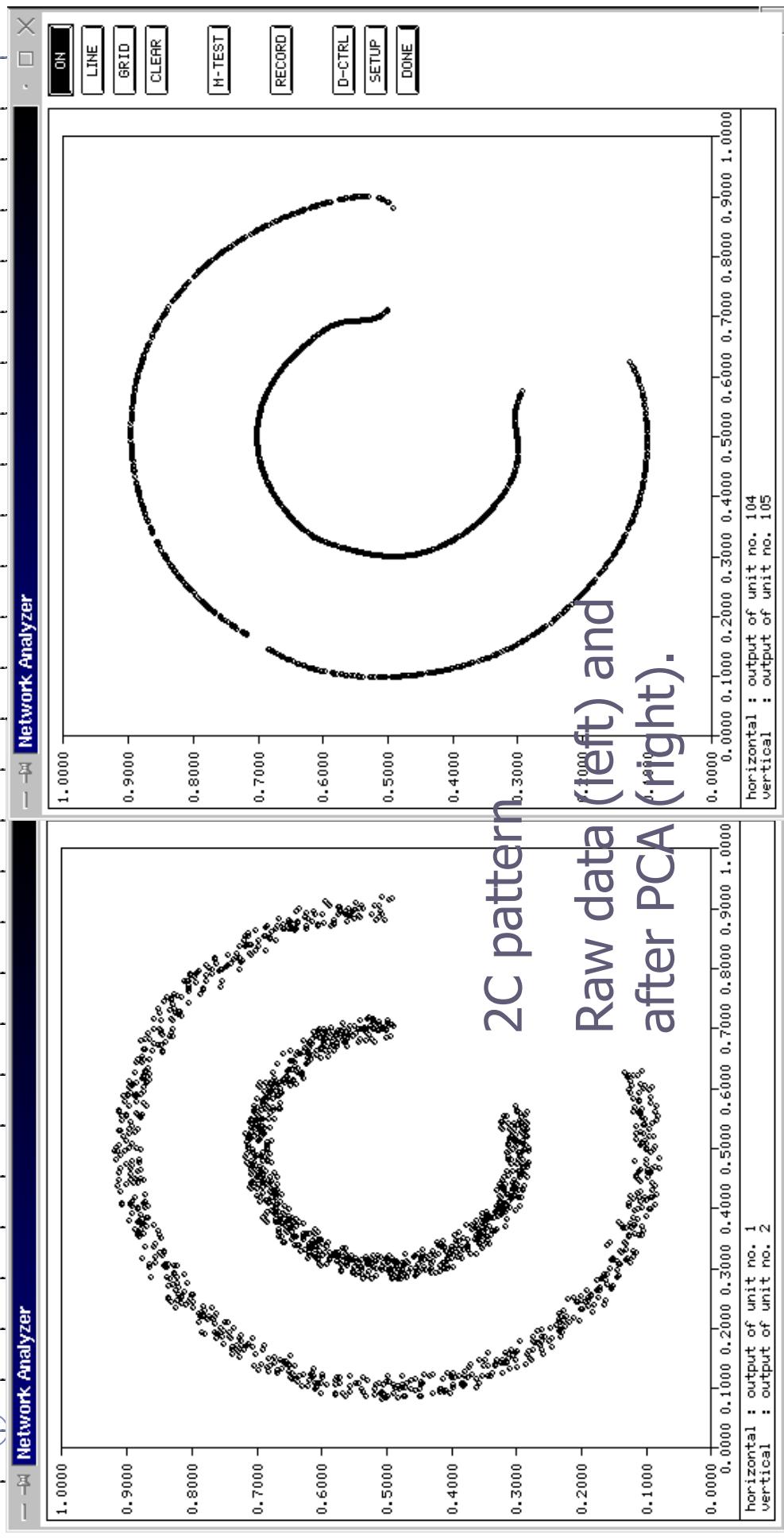
Training a Neural Network



χ^2 as a function of time for training and validation samples.

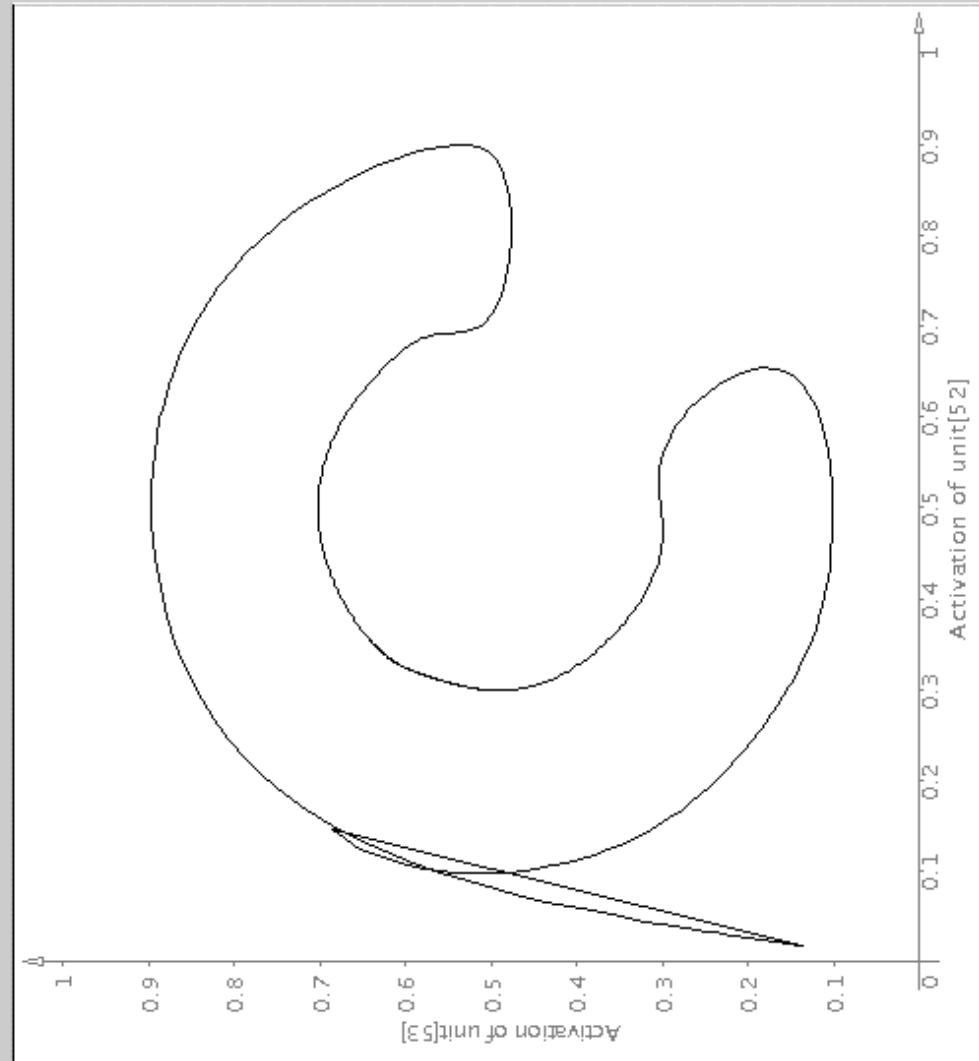
March Wolter, Non-linear PCA with NN

More complicated patterns

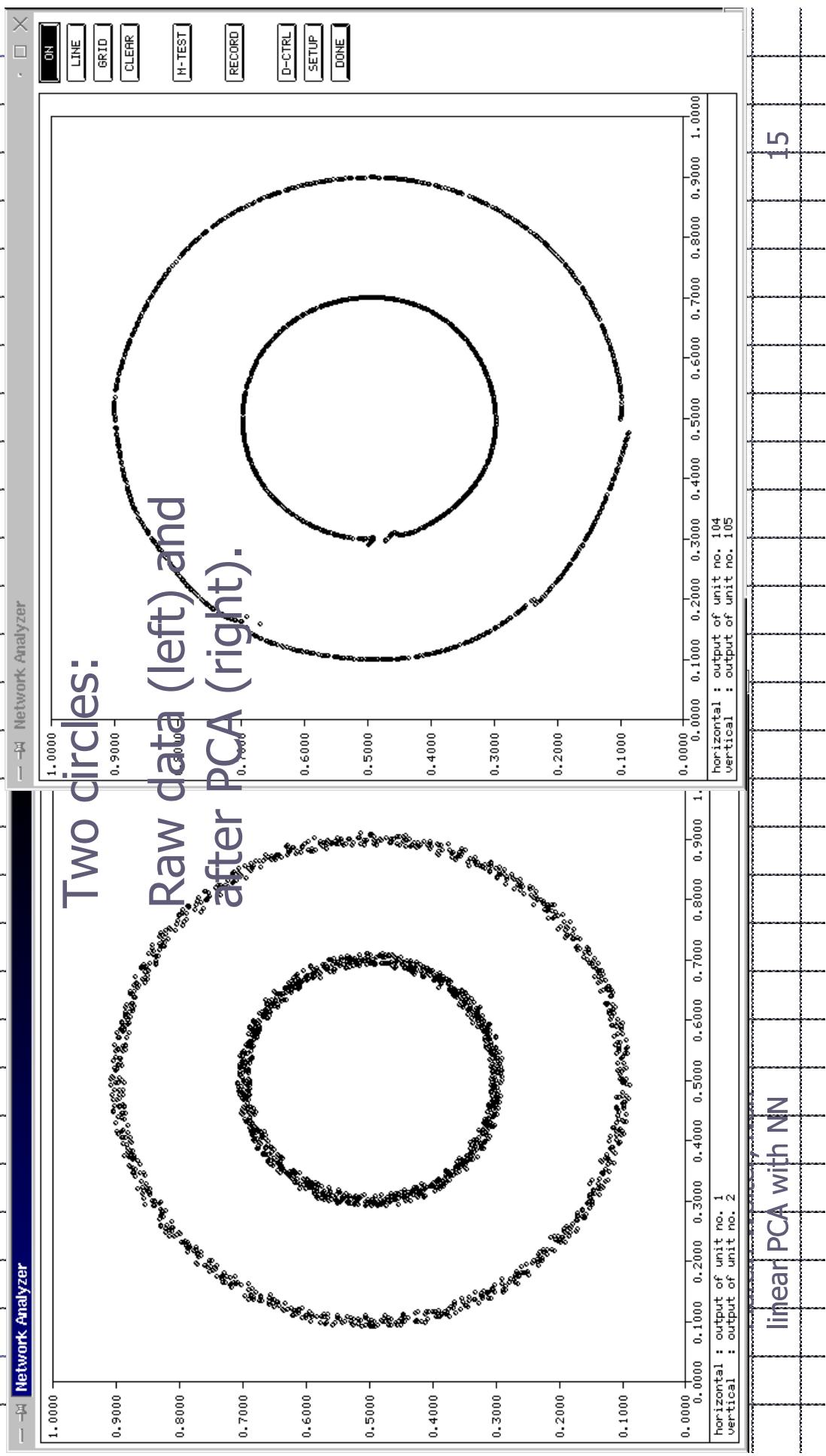


2C pattern

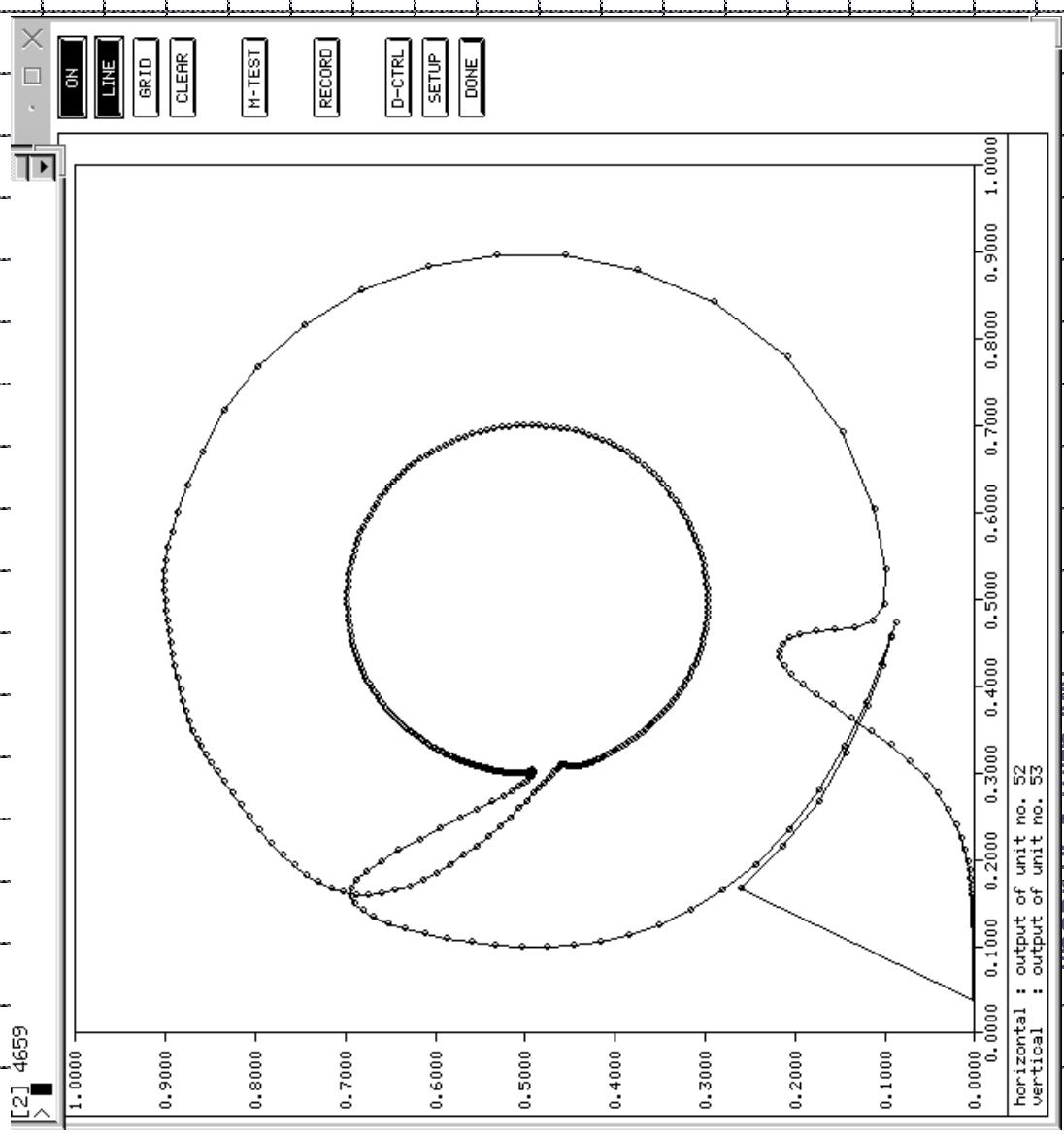
- Transformation of $(-1, 1)$ into 2D using right half of the net.
- "Unused" part of the input range forms an additional, random structure on the left.



More complicated patterns – two circles



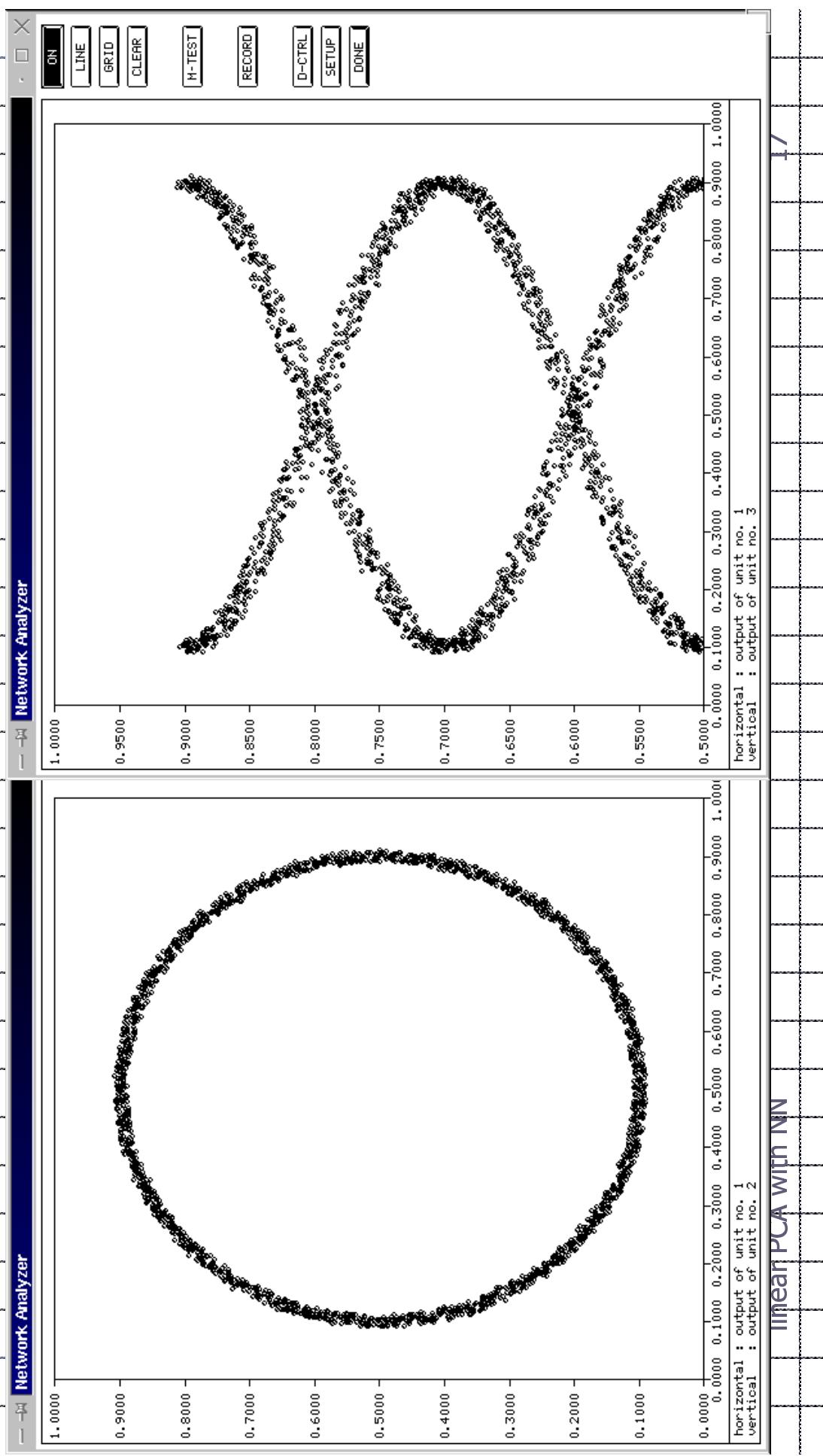
Two circles



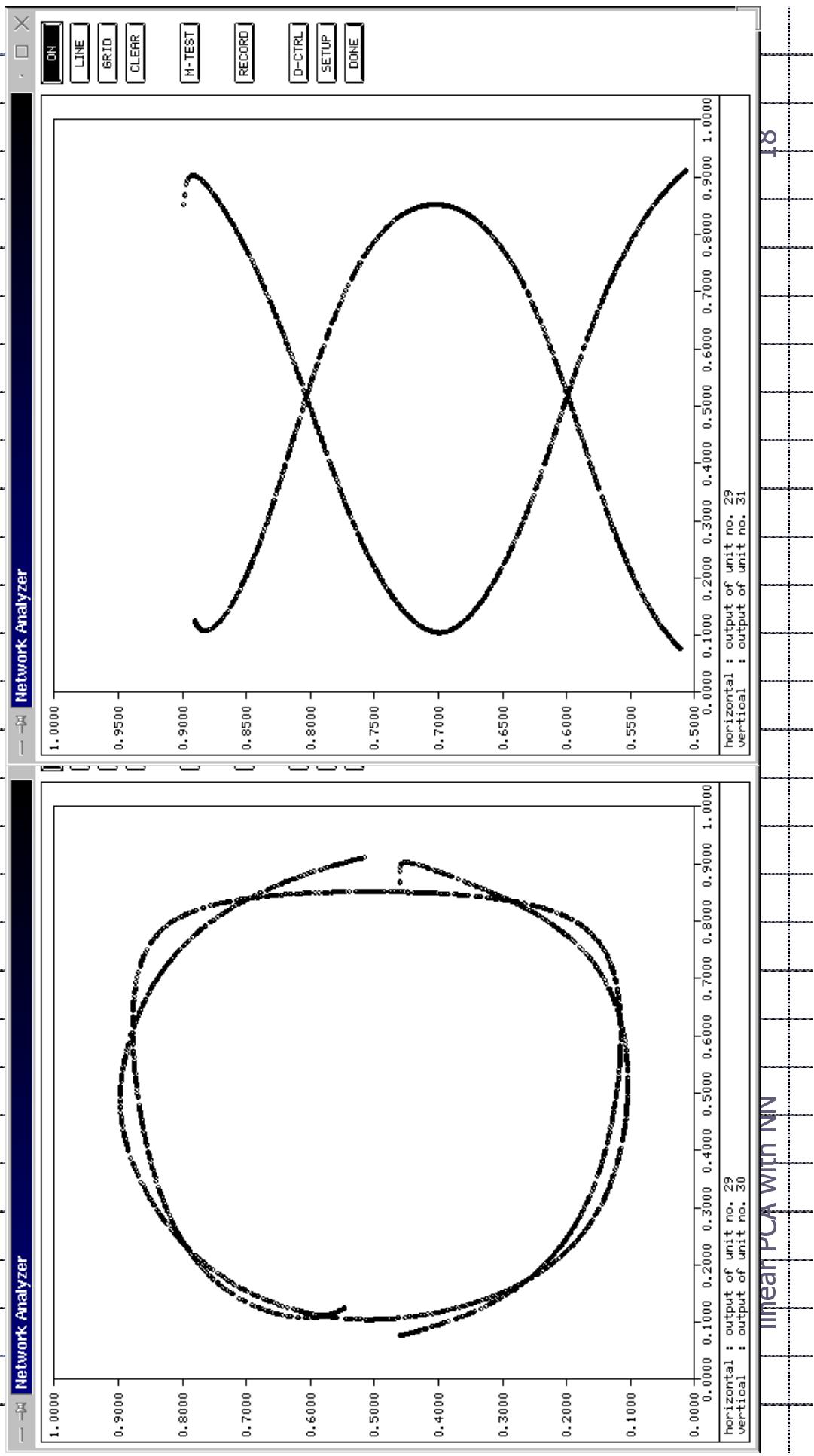
Unused part of the (-1,1) interval contributes to the additional structure on the left side.

TWO spirals - input data

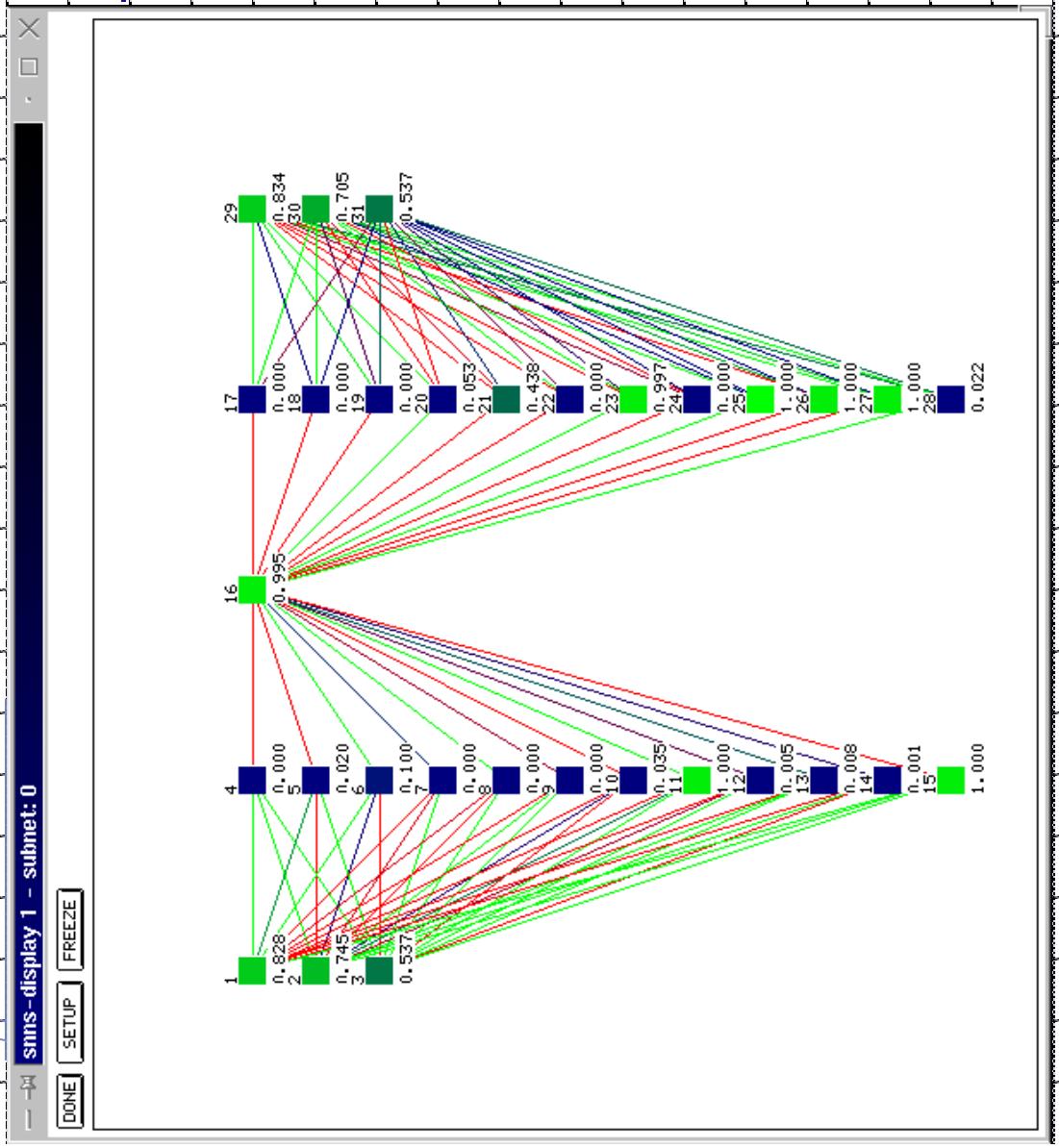
Network Analyzer



Two spirals - after transformation



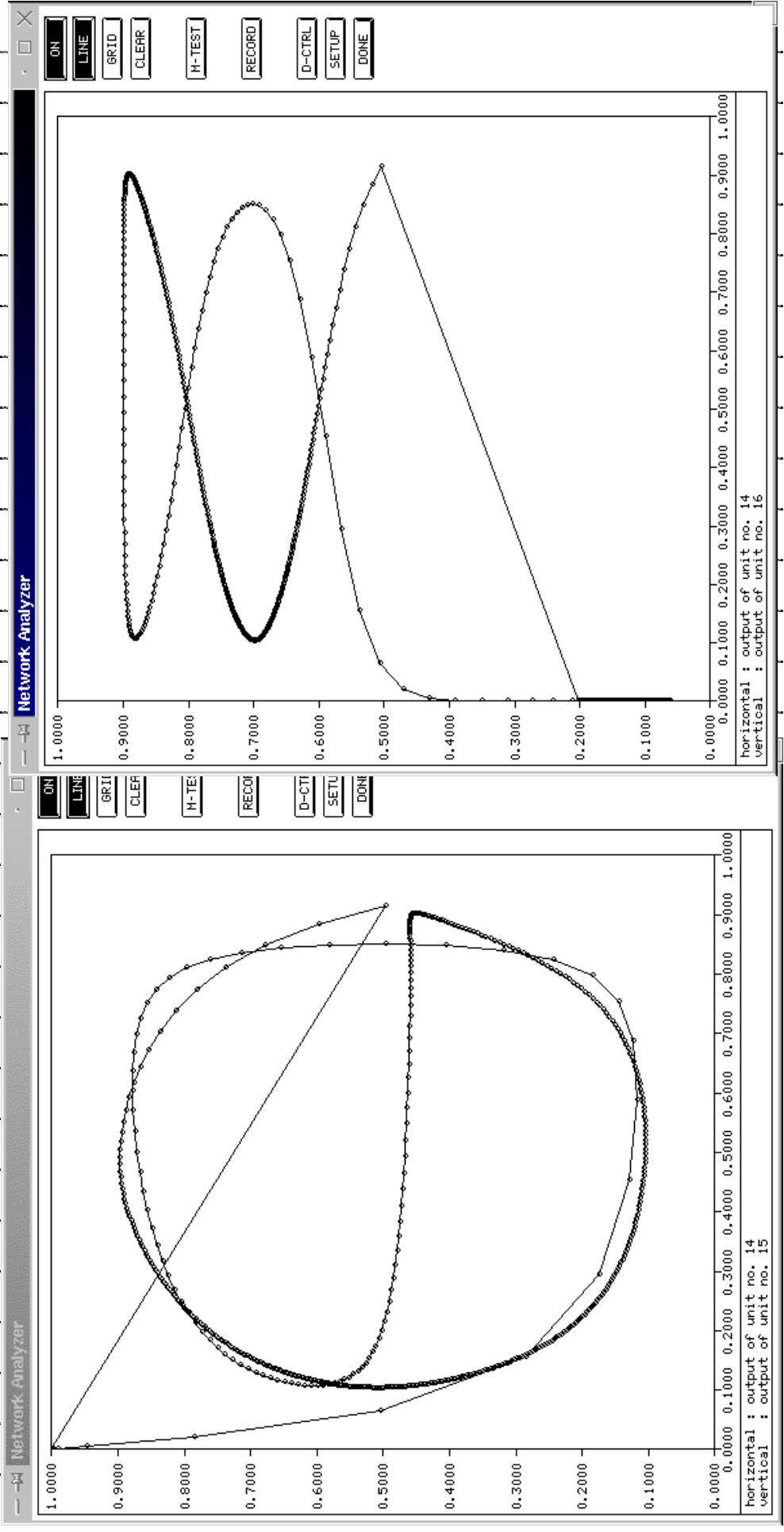
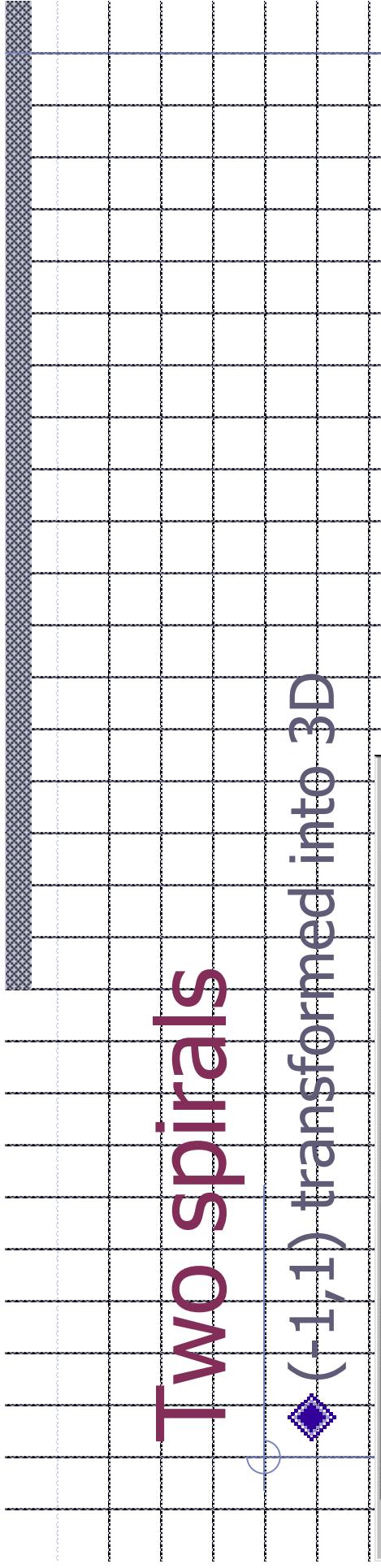
Network used



Network is quite simple – the spiral is not perfectly reproduced

TWO spirals

(-1,1) transformed into 3D



Summary

- ◆ NN can be used not only for pattern recognition but also for a non-linear transformation.
- ◆ It can be effectively used for data preprocessing and dimensionality reduction.
 - ◆ Reduction of input data dimension can significantly improve the quality of pattern recognition (selection).